DOCUMENT CONTROL SHEET

	ORIGINATOR'S REF.		SECURITY CLASS.
	NLR-TP-2003-266		Unclassified

ORGINATOR
National Aerospace Laboratory NLR, Amsterdam, The Netherlands

TITLE
Transforming air transport into a concurrent enterprise
Technical, safety and security perspectives

PRESENTED AT
ICE 2003, Espoo, Finland, 16-18 June 2003

AUTHORS	DATE	PP	REF
E. Kesseler	May 2003	19	13

DESCRIPTORS

Service-based architecture	Federated systems
Concurrent enterprise	Java
Safety	

ABSTRACT
The various parts of the air transport system have evolved independently, with the continuation of the good safety record as overriding concern. Economic realities create an incentive to move from the proprietary stand-alone solutions to an integrated concurrent enterprise solution.
The TALIS case study demonstrates the technical feasibility of the concurrent enterprise for air transport within the required safety levels. Software development processes based on the Unified Modelling Language might not be responsive enough for the required time-to-market for every service. Validated metrics, for use early in the software development, are welcome. A goal-based approach could provide the needed harmonisation of the various safety standards.

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2003-266

Transforming air transport into a concurrent enterprise

Technical, safety and security perspectives

E. Kesseler

This report is based on a presentation held at ICE 2003, Espoo, Finland, 16-18 June 2003.

Customer:	National Aerospace Laboratory NLR
Working Plan number:	I.1.D.2.1
Owner:	National Aerospace Laboratory NLR
Division:	Information and Communication Technology
Distribution:	Unlimited
Classification title:	Unclassified
	May 2003

Approved by author:	Approved by project manager:	Approved by project managing department:
6/6/2003	20030606	4/8/3

Summary

The various parts of the air transport system have evolved independently, with the continuation of the good safety record as overriding concern. Economic realities create an incentive to move from the proprietary stand-alone solutions to an integrated concurrent enterprise solution.
The TALIS case study demonstrates the technical feasibility of the concurrent enterprise for air transport within the required safety levels. Software development processes based on the Unified Modelling Language might not be responsive enough for the required time-to-market for every service. Validated metrics, for use early in the software development, are welcome. A goal-based approach could provide the needed harmonisation of the various safety standards.

Contents

(18 pages in total)

1 Introduction

Historically air transport has taken a safety-first approach. To counteract the inherent dangers of flying, the industry is technology focussed with the continuation of its good safety record as the overriding requirement for all its activities. This has led to an industry where innovation tends to be technology driven. Independent assessment of the safety leads to certification of the equipment, operators, services, etc. used. The result has been specialised, safe, proprietary solutions for the various activities involved. However, these solutions experience a very low rate of innovation and are expensive, compared with the general market. Lack of competition helps to maintain the status quo.

Currently economic pressure, passenger preferences, heightened security concerns and expected long-term traffic growth necessitate a paradigm shift from a technology-driven approach to a customer-oriented or service-oriented approach. This paper describes an industrial case study to determine whether an Internet-enabled service-based architecture could facilitate the transformation to a customer-oriented organisation. As the various services and systems are independently owned and operated, the integration aims for a federated network of co-operating entities. The lessons learned up to now are provided.

As safety remains a prime concern, safety issues are discussed briefly to assess their impact on the architecture. After the tragic September 11, 2001 events, security has become more important. Some preliminary software related results are presented.

2 Existing Theories and Work

Currently the design of a new aircraft, like the Airbus A380, the Boeing Sonic Cruiser (recently transformed to the 7E7) or the Lockheed Martin Joint Strike Fighter, is a major effort which takes well over a decade from initial idea to a flying and certified product. As aircraft will remain in operation for decades and retrofitting equipment is very cumbersome and costly, legacy systems will be inevitable. Even on the ground, Air Traffic Management (ATM) systems take similar times to produce and get operational. In Europe every country has its own, custom made system, and some countries even operate several centres, each with their own dissimilar system. Consequently an integrated air transport solution has to be platform independent as well as take legacy (sub)systems into account.

This situation bears similarities with the US military, which have to integrate many independent systems into a working combination. The US military have initiated the [Joint vision 2020] to transform their existing capabilities into a network-centred enterprise. Combining the platform independence of Java with the networking capabilities of Open wings should, amongst others, enable the Joint Vision 2020.

The software certification standard for aircraft, [DO-178B, 1992] is considered by many as one of the toughest in the software industry. Various parties are currently contemplating the definition of a real-time DO-178B certifiable Java subset plus accompanying virtual machine. The European Space Agency expects it to materialise in the foreseeable future [Claes, 2002], which would satisfy the safety concerns from a technical point of view.

The various aircraft systems are highly integrated to optimise the aircraft flight within the applicable safety limits. However, as aircraft are not connected to air traffic management systems, other then via an old-fashioned voice link between pilot and air traffic controller, this optimisation can not take into account other traffic or other ground system information.

The justification of air traffic management is to prevent collisions between aircraft. Aircraft operate in conditions (e.g. flying through clouds) where the pilots can not do this themselves. Traditionally air traffic management is a national responsibility, where use of civil airspace and airports is optimised to achieve maximum traffic flow while respecting military requirements. The resulting country-specific airspace design combined with full national airspace autonomy, led to systems that are optimised per country.

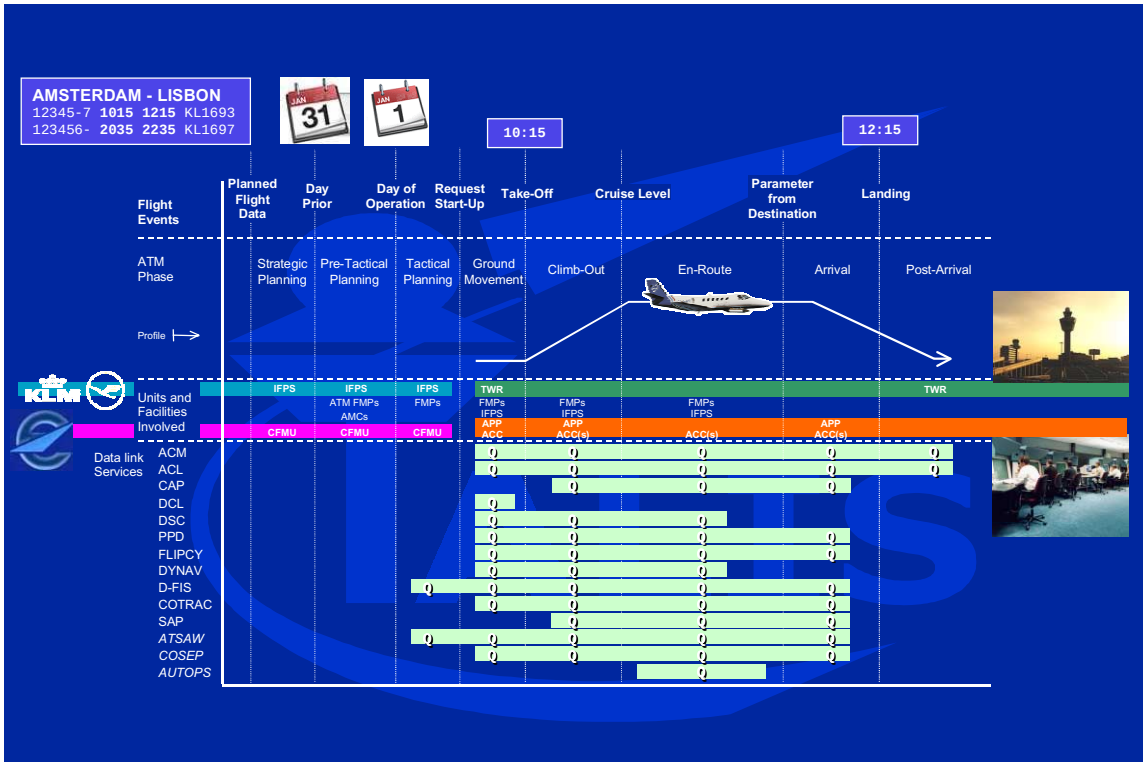

Figure 1 EUROCONTROL collaborative decision making concept and tentative services

Air transport improvement at European level can only be obtained by integrating the existing subsystems in a federated European system. Both EUROCONTROL (the European organisation for air navigation) and the USA Federal Aviation Authority are developing Collaborative Decision Making (CDM) concepts in which all relevant stakeholders like pilots, air traffic controllers, airports and airlines will share information to arrive at user-preferred flights. These concepts define the high-level user requirements. Figure 1 depicts EUROCONTROL's collaborative decision making concept. This concept's high-level objective is to support air traffic controllers, pilots, and all potential ATM users, in all phases of flight by progressively implementing fully seamless communications, data exchange, situational awareness and automation capabilities. Many supporting services are being defined, which are expected to evolve significantly until their planned initial deployment in the 2008-2015 timeframe. That these services are referred to as data link services illustrates the industry's technology-driven approach. After nearly two decades of work on the current, single data link service, it is in only limited pre-operational use, illustrating the need for a dramatically improved time-to-market.

Several Civil Aviation Authorities currently provide much non-critical and consequently non-certifiable information like Notice to Airman (NOTAM), weather reports (TAF, METAR) etc, on the Internet. [Finnish AIS]. Currently this information is not available anymore after the pilots have entered the aircraft, which illustrates some limitations of current proprietary solutions as well as advantages of transforming to concurrent enterprise solutions. Similarly

EUROCONTROL is transforming a lot of airport information to Extensible Mark-up Language (XML), to provide identical and timely information to all actors involved.

Airlines operate in a competitive environment, which might force them into agile software development [Abrahamsson, 2002] for non-safety related services, the extreme opposite from the traditional waterfall model and time-to-market currently used. The current state-of-the-art suggests that the proposed European air transport architecture should be federated, to protect commercial interests, allow the simultaneous support of services with mixed safety and security classifications, be platform independent, deploy Commercial Off-The-Shelf (COTS) to benefit form existing solutions, improve time-to-market, improve affordability and be open to allow for new requirements and solutions.

3 Research approach

To assess the technical feasibility of combining existing (sub)systems into a network-centred service-based architecture the innovative Total Information Sharing for Pilot Situational Awareness Enhanced by Intelligent Systems (TALIS) project has been initiated.

Figure 2 Conceptual view of the TALIS architecture

The TALIS architecture provides an infrastructure to enable the required collaborative decision making concepts. Within the project, the architecture will be developed and a prototype implemented with two demonstration services. Figure 2 depicts the TALIS architecture.
To describe how the air transport concurrent enterprise concept would work, the sample TALIS service depicted in figure 3 is described. The sample service focuses on a pilot user. On an airport the pilot's information needs differ depending on the flight phase. A co-ordinated pushback service will allow the pilot to improve the reliability of on-time pushback. For this the pilot needs amalgamated information from fuelling services, baggage-handling services, catering services, security services, gate personnel, Airline Operations Centre (AOC) for information on connecting passengers etc. A co-ordinated pushback service optimises usage of the taxiway linking the various gates and prevents aircraft from blocking each other. Subsequently Surface Movement Guidance and Control System (SMGCS)-based taxi services

guide the aircraft to the correct runway, optimised for other airfield traffic and taking possibly adverse weather or airfield maintenance restrictions into account. Finally runway incursion alerting services, using surveillance services, improve the safety during take-off. For arriving aircraft taxi-services guide the aircraft and the ground handling vehicles to the (re) allocated gates. Such TALIS powered services illustrate the power of integrating existing services i.e. deploy the concurrent enterprise.

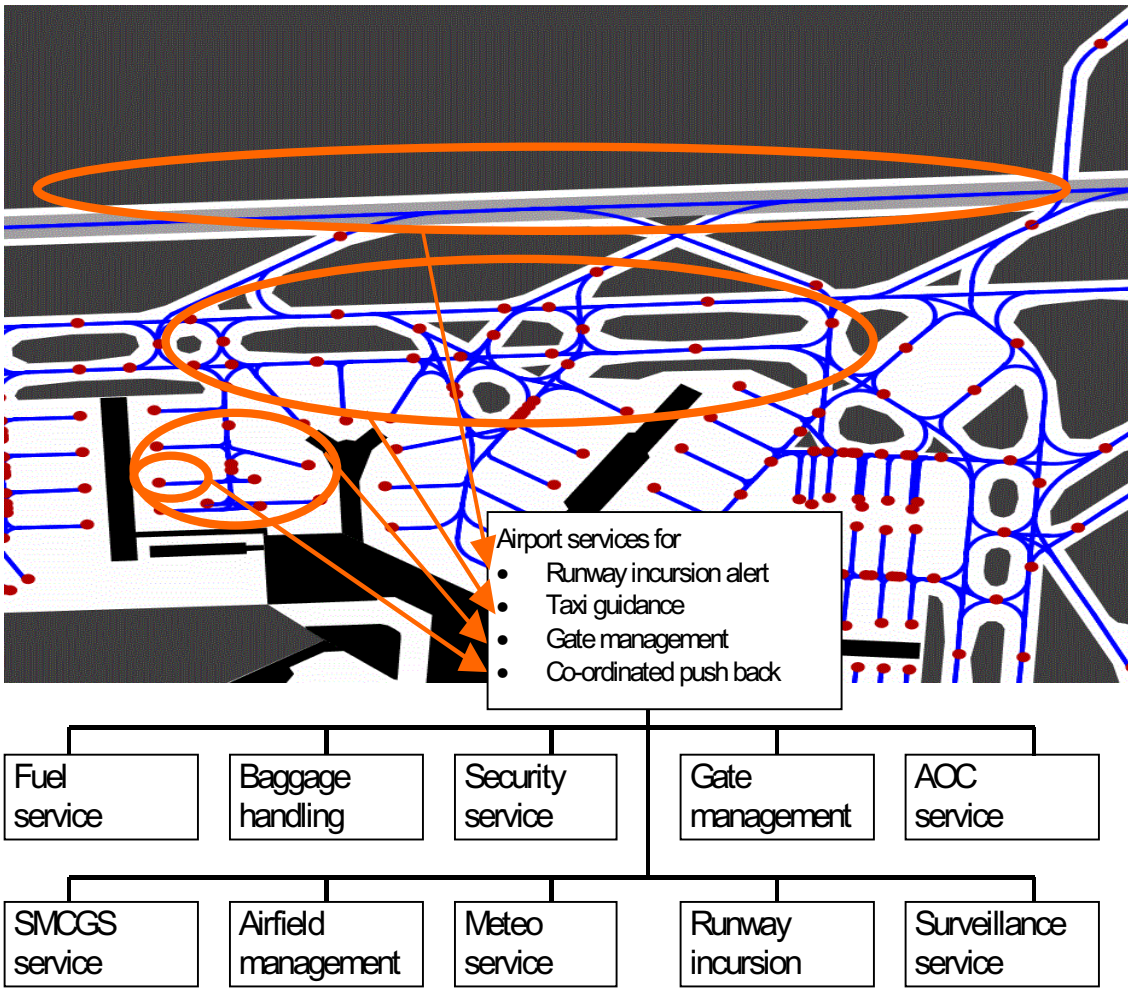

Figure 3 Sample TALIS airport service

Java's platform independence supports uplinking new data or even new software to the aircraft. This facilitates a swift deployment of updated or even new services, also for aircraft with legacy avionics. This big advantage is already commonplace for concurrent enterprises in other domains. Consequently TALIS will be Java-based.

Summarising, the major TALIS requirements include:

- Build upon the many disparate legacy systems in use and take into account the various independent (sub)system owners (federated architecture)
- Be flexible by configuring the architecture to the needs of the local user, e.g., pilot, air traffic controller, airline, gate manager, fuel service, luggage handling, meteorological offices, etc. This is referred to as variability in software product lines (variability)
- Allow interconnection with the various, non-harmonised existing systems of the wide variety of actors (cross platform connectivity)
- Provide plug-and-play at system level to allow users and their services to enter and leave the network at their own discretion without interrupting the other provided services (self-forming self-healing network)
- Provide simultaneous support of both services with a safety impact as well as non-certifiable services (multiple-level safety support)
- Support simultaneously many communication technologies, which will evolve with commercial speeds. The architecture should survive the specific supporting technology, e.g., Aeronautical Telecommunication Network (ATN), UMTS, satellite communication, wireless LAN, bluetooth, etc. (technology independence)
- Foster innovation by providing an open system to support new, as yet unknown, services to improve the responsiveness of the air transport system (open system)
- Be affordable by deploying COTS solutions wherever possible. Open systems also facilitate competition which further improves the affordability (affordable)
- Improve time-to-market by deploying COTS and concentrate on air-transport specific parts (time-to-market)
- Provide simultaneous support of both services with a security impact as well as services without security concerns. (multiple-level security support)

4 TALIS architecture

Based on the challenge described above, the TALIS architecture provides the middleware to integrate the existing elements. By combining the strengths of the individually provided services the time-to-market for new services can be reduced significantly and competitiveness increased resulting in better service at lower costs. TALIS has chosen to exploit COTS as much as possible, e.g. the Internet Protocol (IP) to provide cross-platform connectivity, Java to provide cross-platform applications and OpenWings to add the plug-and-play (self forming, self healing) capability at system level. Java also provides code mobility, enabling the swift update of services needed by the concurrent enterprise approach to air traffic service provision. What remains for TALIS is to define the architecture, the application specific service interfaces and the supporting services around the components. These components can be new components or legacy systems with a wrapper. TALIS also needs to address the safety and security requirements specific to air transport. Table 1 summarises the TALIS requirements are their compliance.

Table 1 Summary of TALIS requirements compliance

Requirement	**Compliance**
Federated architecture	OpenWings
Variability	Java
Cross platform connectivity	Java + OpenWings
Self-forming self-healing network	OpenWings
Multiple-level safety support	TALIS
Technology independence	TALIS (specific protocols) + OpenWings
Open system	TALIS + Java
Affordable	COTS
Time-to-market	TALIS + COTS
Multiple-level security support	TALIS + COTS

5 Process observations

This section provides the lessons learned while applying the concurrent enterprise to air transport. In air transport traditionally structured methods are used combined with a waterfall life cycle. Combined with company-specific software processes these lead to safe and certifiable products but at comparative high cost, long time-to-market and limited variability. To improve on this, TALIS has chosen the Unified Modelling Language (UML) design method and the Unified Software Development Process (USDP) development process, a major innovation for this industry. Theoretical knowledge of UML and USDP was available at some partners but all were lacking practical experience.

Many of the TALIS requirements need to be accommodated in its architecture. Consequently the quality of the architecture becomes of paramount importance. The Rational Rose UML tool suite provides much fewer model checks then the tools, with proprietary extensions, we were used to. Consequently the planned formal reviews became more important. The reviews were held in accordance with European Space Agency (ESA) [ECSS-E-40, 1999] practises. An improvement would be to use a documented assessment method like Scenario-Based Analysis of Software Architecture (SAAM) [Kazman, 2003] during such review. The disadvantages of any review are the high cost and non-repeatable results. Consequently feedback on architecture decisions is only provided at a few moments, when a lot of work has already been done. Automatically generated metrics, which provide immediate feedback, have proven their worth with structured methods. Some trials were performed, but the tools used presumed UML model conventions. These conventions turned out to be incompatible with the project conventions, which are based on available company practises and document generation tools. The correct way for metrics would be to use the Goal Question Metric approach [Basili, 1994], but only one validated metric for statecharts was found. Experience with automatic architecture metrics [Chaudron, 2003] tends to suggest that after manual interpretation they may point to design weaknesses. Most benefit is acquired when watching the trend in these metrics during the design period. More automated and validated metrics that provide direct feedback during the architecture design are welcomed.

The implementation of the TALIS prototype architecture is on track. A demonstration is expected by the beginning of 2004. Exploiting COTS already proved its worth, since the start of the project Jini has been superseded by OpenWings which provides much better technology independence. Also work on a DO-178B certifiable Java subset has started in the Open Group [Foss, march 2003]. Once completed this will allow the TALIS architecture to support any application, irrespective of its safety criticality.

Preliminary results for a 2 KLoC (thousand Lines of Code) component to display an aircraft instrument (the safety critical primary flight display) of the 80 KLoC cockpit display subsystem indicate that Java halves the implementation time with respect to the current Ada and C based

practise. Re-use, amongst others of graphic components, is the major contributor to this improvement. In case of certification, improvement is also expected for module tests and test effort.

Using the USDP process at project start three iterations of a traffic information service were foreseen with two iterations for the meteorological application and only one iteration for the architecture. It is important to note that for certifiable software much effort is needed for verification and validation, both of which have been excluded from the TALIS applications. The economic downturn resulted in a major effort shift between partners for the latter application, causing a reduction to one iteration and a minor project extension. For the former application the UML and USDP learning curve caused a reduction to two iterations. UML and USDP considerably reduce to time-to-market with respect to current practise. Nevertheless for services without safety implications and stringent time-to-market requirements another more responsive software paradigm e.g. from agile software development needs to be considered.

6 Safety and security issues

Relying exclusively on certified products, services, procedures and people ensures air transport safety. For aircraft, certification is performed once before entry into service. For every subsequent modification an additional certification is needed. For ground systems and maintenance a license is provided for a fixed period of time. After expiry another assessment is performed before extending the license period. Additionally all operators, such as pilots, air traffic controllers and maintenance personnel need personal licenses. These licenses also cover a fixed time period with checks upon renewal. For all parts of the air transport system incident and accident reports are produced and analysed to continuously improve all elements concerned. These procedures have been so successful that accident rates, in the developed world, continue to drop and for all accidents the human instead of technology is now the major contributing factor by far.

As TALIS integrates aircraft, ground systems and some services, many different safety standards apply for its components. Due to its different background each standard evolved differently and now imposes, with due justification, specific not harmonised requirements. As various services will be of different safety criticality, the architecture supports this. The TALIS architecture itself complies with the requirements of the most critical service it intends to support.

As software cannot be tested or be proven to comply with the required failure rates of once per billion operating hours for the most hazardous class, all standards impose requirements on the software development processes. The various standards impose different life cycle models with different software artefacts so harmonisation or mutual recognition is needed in order not to impede market conformant time-to-market and service update rates. Also current standards are lagging behind modern technology, e.g. object orientation and COTS are still not accommodated. This delay incurs lot of additional effort on suppliers wishing to use such new technologies. In a goal-based approach like [SW01, 1998] the software is classified, the required evidence defined and a reasoning needs to be provided that the evidence satisfies the safety requirements. This approach is more appropriate for a federated system then the many incompatible standards currently prescribed.

Systematic application of security in air transport systems is innovative. The use of special small-volume technology used to work as a kind of deterrent. This deterrent is already not effective anymore. TALIS reliance on COTS both necessitates security measures as well as facilitates the deployment of existing solutions. In line with the TALIS philosophy security is best addressed by adhering to an open, internationally recognised standard. The [Common Criteria 1999] originating from the military domain, provide objective evidence about the product security level. Qualified and officially recognised assessors perform the objective and repeatable evaluation, much like for safety certification. The common criteria impose software

process requirements as well as a standard list of security functions from which to chose. Further study is needed to check whether the common criteria address all security concerns and assess the compatibility of yet another set of software process requirements.

7 Conclusions

The TALIS prototyping work demonstrates that it is technically feasible to transform the air transport system from the current set of stand-alone proprietary systems to a service-based architecture enabling a concurrent enterprise. The federated architecture allows current legacy systems to be incorporated. The extensive use of COTS improves affordability, time-to-market and competition, and reduces obsolescence by continuously incorporating new supporting technology. The result would be a more responsive air transport system

UML and USDP improve the time-to-market. The lack of validated automatically generated metrics makes intermediate assessment of the architecture cumbersome. For services without safety or security impact but stringent time-to-market requirements, other software paradigms, like agile software development, might be more appropriate.

The proposed TALIS architecture and its supporting COTS technology can simultaneously provide services of various safety levels. Even the most safety critical services could be accommodated after completion of market-driven progress on Java. A goal-based approach is recommended to harmonise the various, currently incompatible, (sub)system certification standards.

For security concerns, which arose after the project start, a promising standard and certification scheme has been identified. Its impact has to be studied further.

Acknowledgement

This work has been partly funded by the European Commission, DG Information Society Technologies as project IST-2000/28744. The EUROCONTROL Experimental Centre leads the consortium of Lido, NLR, Thales Avionics and Skysoft.

References

Pekka Abrahamsson: Agile software development methods: A mini-tutorial. WWW page. http://agile.vtt.fi/seminar2002.html, accessed March 2003.

V.R. Basili; C Caldiera; H. D. Rombach: Goal Question metric paradogm. Encyclopedia of software engineering, Volume 1, John Wiley & Sons, 1994, p.528-532.

Michel Chaudron; Johan Muskens: A comparative study of software architecture metrics in embedded and information system. WWW page. http://metric.cse.unsw.edu.au/Metrics2003, accessed March 2003

Peter Claes: The Galileo SW Development Context - Risks, Context, Ground Segment engineering, Space Segment engineering, PA/QA, Safety, and SW Standards. WWW page. http://www.estec.esa.nl/conferences/02c30/index.html,

Common criteria for security evaluation, Version 2.1. WWW page. http://www.commoncriteria.org/cc/cc.html, August 1999, accessed March 2002. Also know as ISO/IEC 15408

DO-178B / ED12BL: Software Considerations in Airborne Systems and Equipment Certification. RTCA & EUROCAE, December 1992.

European Co-operation for Space Standardisation (ECSS). WWW page. http://www.ecss.nl/, accessed March 2003.

Finnish Aeronautical Information Service (AIS),

- Pre-flight Information Bulletins (PIB). WWW page. http://www.ilmailulaitos.fi/bulletins/Bulletinvalikko.htm
- Notice to Airman (NOTAM) summary. WWW page. http://www.fcaa.fi/bulletins/summary/notam_summary.pdf, accessed March 2003.

Jim Alves Foss; et al: Partitioning kernel protection profile, version 1.2. WWW page. http://www.opengroup.org, accessed March 2003 (members only).

Joint vision 2020. WWW page. http://www.dtic.mil/jv2020/jvpub2.htm, accessed March 2003.

Rick Kazman; G. Abowd; L. Bass; Clements: Scenario-based analysis of software architecture . WWW page. http://www.sei.cmu.edu/architecture/scenario_paper/ieee-sw3.htm , accessed March 2003.

SW01: CAP 670 ATS Safety Requirements, UK CAA. WWW page. http://www.caa.co.uk/docs/33/CAP670.pdf , April 1998, accessed March 2003.

Acronyms

AOC	Airline Operations Centre
ATM	Air Traffic Management
ATN	Aeronautical Telecommunication Network
CDM	Collaborative Decision Making
COTS	Commercial off-the-Shelf
IP	Internet Protocol
KLoC	Thousand Lines of code
NOTAM	Notice to Airman
SMGCS	Surface Movement Guidance and Control System
SAAM	Scenario-Based Analysis of Software Architecture
TALIS	Total Information Sharing for Pilot Situational Awareness Enhanced by Intelligent Systems
UML	Unified Modelling Language
USDP	Unified Software Development Process
XML	Extensible Mark-up Language